Lighting Your Way Home

Finding the Optimal Path to the MBTA

Sarah Ferry, Dimitri Makrigiorgos, Alex O'Connor
ferrys@bu.edu, dmak1112@bu.edu, aoconno8@bu.edu

1. Introduction

Going to an establishment that allows alcohol consumption on the premises can be fun, but getting home can be risky. Driving after drinking is extremely dangerous, and even getting to nearby public transportation stations can present risks, especially if the routes to those public transportation stations are not in well lit areas. After spending four years at Boston University, each member of our group had intimately experienced danger trying to get home at night, so we felt motivated to do something to alleviate this issue. For our project, we decided to tackle a small part of the problem by mapping routes from every location in Boston with an alcohol license to an optimal MBTA stop, using distance, number of streetlights, and variance of streetlights as variables. By doing this, we hope to take a step towards allowing the citizens of Boston to have a fun night out without having to worry as much about how to get home safe.

2. Data

2.1. Data Sources

2.1.1. Section 12 Alcohol Licenses: This dataset from Analyze Boston provided us with the locations and names of all of the establishments that are allowed to sell alcohol on their premises.

2.1.2. Streetlight Locations: This dataset from Analyze Boston provided us with the latitude-longitude locations of all of the streetlights in Boston.

2.1.3. MBTA V3 API developer portal: This data source provided us with the name and location of all of the MBTA stops in Boston.

2.1.4. OpenStreetMap: This data source provided us with information about the entire street map of Boston. We accessed OpenStreetMap via OSMnx, which is a Python package that allows for access to the OpenStreetMap API.

2.2. Collection and Preprocessing

In order to perform useful algorithms on these datasets, we performed transformations to combine them in a way that would enable us to find the optimal paths.

Initially, we found the three closest MBTA stops to each alcohol license establishment. In order to determine the distance between each MBTA stop and establishment, we needed to acquire the address of the establishment from the Section 12 Alcohol Licenses dataset and use the Google Maps Geocoder to determine the latitude and longitude point. We then used an R-tree index to efficiently find the three nearest MBTA stops.

Using this dataset, we determined a radius around each establishment that was equal to the distance from each establishment to the third nearest MBTA stop. We then incorporated the streetlight dataset to produce a dataset that contained each alcohol establishment, the three nearest MBTA stops, and all of the streetlights within the radius that we created. The purpose of this was to be able to calculate the number of streetlights along the routes in order to optimize the overall route safety.

3. Optimization

Our goal was to find the optimal route from an alcohol establishment to an MBTA stop. We used the dataset we created during preprocessing to calculate the optimal path from each alcohol license an MBTA stop. To accomplish this, we generated a driving map of the city of Boston using the OpenStreetMap API OSMnx and found the closest node on this map for each of the alcohol establishments, MBTA stops, and streetlights. We then calculated both the shortest paths and the safest paths from each alcohol license to its three closest MBTA stops using the shortest path method from the NetworkX Python package. We determined the safest path to be the path that includes the node with the most streetlights in the radius.

Finally, we performed an optimization technique that scored all six of the paths based on a number of factors including route distance, number of

streetlights, and variance of streetlights along the path. By finding the highest scored path, we hoped to be able to provide civilians with routes to nearby MBTA stops that provide a reasonable combination of convenience and safety.

3.1 Scoring Method

Our scoring method takes the paths generated from each alcohol license (safest and shortest path to nearby MBTA stops) and scores them based on variance of streetlights on the path, total number of streetlights, and distance. The equations are as follows:

- Shortest path score = $.4v + .4n - .1d$
- Safest path score = $.45v + .4n - .15d$

where v is the variance of streetlights on route, n is the number of streetlights on the route, and d is the route distance.

Since we know that the safest path is either longer or the same as the shortest path, we weigh the distance less on the safest path and the variance a bit more. This is because the safest path contains the node with the most streetlights, but we want to ensure that its streetlights are well distributed throughout the entire route.

4. Statistical Analyses

For our statistical analyses, we tested two hypotheses and checked for correlation with the hope of discovering interesting patterns within our determined routes. The first hypothesis that we tested was to determine if the mean number of streetlights at the starting node of a route is greater than the mean number of streetlights at the ending node.

- H_O: μ(starting node) - μ(ending node) > 0
- H_A: μ(starting node) - μ(ending node) <= 0
- α = .01

The resulting z value was 5.70 and p value was less than .0001. With a p value of <.01, we are able to reject the null hypothesis and conclude that there is sufficient evidence at α = .01. Therefore, the mean number of streetlights at the ending node (MBTA stop) of a route is greater than or equal to the mean number of streetlights at the starting node (alcohol license).

The second hypothesis we tested was to see if the mean number of streetlights at the start and end nodes of a route was greater than the mean number of streetlights at all of the middle nodes in the route.

- H_O: μ(start + end) - μ(middle) > 0
- H_A: μ(start + end) - μ(middle) <= 0
- α = .01

The resulting z value was -6.14 and p value was .9998. With a p value that is much greater than .01, we are unable to reject the null hypothesis, meaning that there is not sufficient evidence at α = .01. Therefore, the mean number of streetlights at the starting and ending nodes in a route is greater than the mean number of streetlights at all middle nodes in the route. This result is one of the reasons why our project is so important: while well established places like restaurants and MBTA stops are generally well lit, the routes in between them generally are not. While this result may seem obvious, it provides even further justification for incorporating the variance of streetlights into our optimization score.

Additionally, we ran three tests to check for correlation:

1. Number of streetlights at starting node vs ending node, r = .2007
2. Number of streetlights at ending nodes vs middle nodes, r = .4792
3. Number of nodes in a route vs total distance of route, r = .1683

Since the correlation coefficient for all three of these tests was close to zero, all three tests for correlation did not yield convincing results. None of these tests show strong correlation between the factors.

Finally, we determined the average number of nodes in a route and the average distance of a route. We found that the average number of nodes was 3.9375, and the average distance was 525.008 meters. This indicates that in general, our routes were walkable, which is a very important feature of our analysis.

5. Visualizations

The last step in our process was to visualize the data, which we did in two ways.

First, we used the python package Folium and the output dataset, which contained the optimal routes from each establishment to a nearby MBTA stop, to create a mapped out version of the data. This resulted in an interactive map of Boston, where each establishment has an optimal path to a nearby MBTA stop. Clicking on any of those routes will display information about the route, including the name of the establishment, the name of the MBTA stop, the number of streetlights on the route, and the total length of the route in meters (see Figure 1). Additionally, clicking on the starting node or ending node will display the name of the establishment.

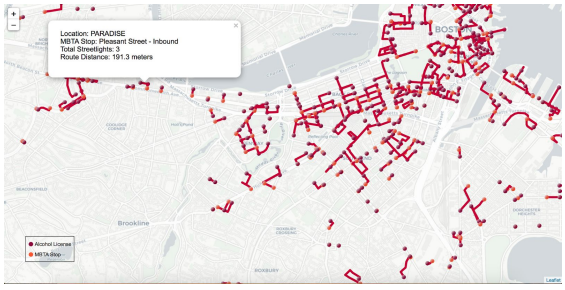

Figure 1: Folium visualization of the optimal routes

For our second visualization, we used d3 along with a dataset that contained the score and distance of each route to create two interactive graphs. The first shows how the distances of each route vary as the score increases between the safest routes and the shortest routes (Figure 2), while the second tracks the same interaction for only the optimized routes (Figure 3). We were able to observe visual trends from the graphs, such as the fact that distance tends to decrease as score increases, which results from the fact that our scoring algorithm attempts to minimize distance of each route while still maintaining a reasonable number of streetlights along the path. Differences can also be seen between the safest and shortest route lines, as the safest route score is less influenced by distance than the shortest route score.

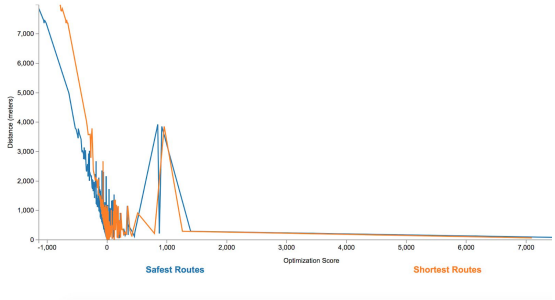

Figure 2: Distance vs. Optimization Score for Safest and Shortest Routes

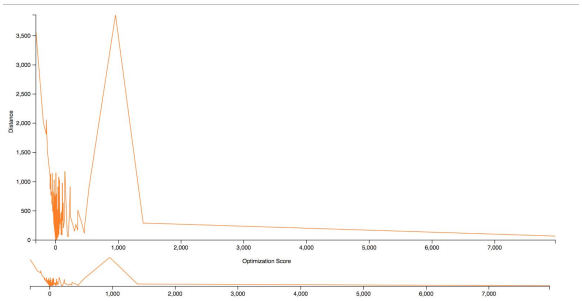

Figure 3: Distance vs. Optimization Score for Optimal Routes

Finally, we created a Flask web service to display our visualizations. The basic app allows a user to toggle between our three visualizations using HTML buttons.

6. Tools and Techniques

Since the different packages we used required data in specific formats, we encountered some situations where we had to transform or adjust the data to be able to effectively use the tools.

To begin with, we had large amounts of data and were performing complex transformations, so our algorithms could take an exceedingly long time. As a result, we had to use tools to improve the running time, like an R-tree index.

6.1 Tools

We used the following tools to assist us with calculations. We encountered some interesting issues, necessary transformations, or limitations along the way with each tool.

6.1.1 OpenStreetMap, OSMnx, & NetworkX: Initially, we started the project wanting to use a collection of sidewalk data from Analyze Boston, so that we could create the shortest walking paths. This data was in GeoJSON form and contained many unconnected lines that mapped out the sidewalks in Boston. We decided that it was out of the scope of this project to create a connected graph out of this ourselves, so we used OpenStreetMap via OSMnx. The street data from OSMnx is in the form of nodes and edges, which presented a limitation: we were forced to find the nearest nodes to our longitude and latitude points, as opposed to actual longitude and longitude points. This resulted in a loss of precision which affected the results of our project. Additionally, since we used OpenStreetMap's driving map, we occasionally ran into a problem where the route was extremely long due to accidentally getting stuck on the highway. To make our visualization more useful, we removed those outliers.

6.1.2 Scipy & Statsmodels: No issues arose from using Scipy and Statsmodels, as the tests and packages were fairly straightforward. The only steps that had to be made were for formatting purposes (so that the packages accepted the data) and minor calculations. For example, since our statistical analyses tested averages of streetlights across all routes, we first had to sum the number of streetlights per route and then perform the analyses.

6.1.3 d3: The multiple versions of d3 presented issues while creating visualizations of our data. In addition to re-working our data to a ".csv" file for input, we encountered problems with zooming and toggle functionality on the same graph. Spacing, scale, and labeling presented problems as well. We settled on using two different visualizations using d3, one that represents the optimized routes and using zooming and brushing tools, and one that contrasts the safest and shortest routes, which uses the toggle tool.

6.1.4 Folium: We used OSMnx and Folium generate a Leaflet interactive map to display the route data. Our final code to produce the map uses an adaptation of the code provided in OSMnx to generate Folium visualizations, since it did not directly fit our use case.

6.1.5 geoql: Geoql only accepts data in the format of GeoJson data, so we had to transform our streetlight data into GeoJson to be able to use the geoql library to only retain streetlights within the specified radius. We also noted that the latitude and longitude coordinates were flipped in the geoql library, so before we realized this we had issues getting the proper coordinates of each streetlight.

7. Conclusion & Future Directions

We hope that by creating this information and these visualizations detailing safer paths from establishments that serve alcohol to MBTA stops, we can help decrease the risk of getting home after consuming alcohol. Obviously, this will not eliminate all risk associated with drinking and getting home, but we believe this is a step in the right direction towards making Boston a safer neighborhood. Throughout the process, we noticed opportunities where we could expand the project but were not able to pursue them all. If we had more time, we would make the visualization aspect more interactive, so that a user would be able to enter in a specific establishment and the MBTA stop that they were interested in going to, and have the path generated for them dynamically. We were also limited in the routes that we made because the data we used to create the routes was street data and not sidewalk data. Ideally, we would have created a graph of the sidewalk data so that we could calculate actual walking directions from each establishment to MBTA stop. Additionally, we would aim to discover more interesting statistical analyses and visualizations related to this type of project, since ours were not extremely useful in furthering our results. Finally, we would include other features to help people get home safe, possibly incorporating information such as using Uber.

8. Citations

1. Department of Innovation and Technology, "All Section 12 Alcohol Licenses", Analyze

Boston [Data set] (Accessed April 22, 2018).
2. Department of Innovation and Technology, “Streetlight Locations”, Analyze Boston [Data set] (Accessed April 22, 2018).
3. MassDOT Developers , “MBTA V3 API Portal” [Database] (Accessed April 22, 2018).
4. © OpenStreetMap contributors, “OpenStreetMap”, OpenStreetMap Foundation, [Database] (Accessed April 22, 2018).
5. GoogleMaps APIs, “GoogleMaps Geocoding API”, Google (Accessed April 22, 2018).